

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

1. (Currently amended) A method for detecting whether a routine has stalled, comprising:

accessing existing code for a first routine;

automatically modifying said existing code by changing said existing code and adding new code; and

using said new code and said changed existing code to determine if said first routine has stalled, wherein said step of automatically modifying comprises:

adding object code for a timing mechanism to said existing code,

adding a first instruction to said existing code for said first routine, said first instruction causes a starting of said timing mechanism for said first routine, said first instruction is object code, and

adding a second instruction to said existing object code for said first routine, said second instruction causes a stopping of said timing mechanism for said first routine, said second instruction is object code.

2. (previously presented) A method according to claim 1, wherein:

said existing code is modified prior to starting execution of said existing code;

said existing code is object code; and

said new code is object code.

3. (Currently amended) A method according to claim 2, for detecting whether a routine has stalled, comprising:

accessing existing code for a first routine;

automatically modifying said existing code by changing said existing code and adding new code; and

using said new code and said changed existing code to determine if said first routine has stalled, wherein said existing code is modified prior to starting execution of said existing code, said existing code is object code and said new code is object code; and

further comprising:

receiving a user created rule, said rule identifies said first routine and an interval, said step of automatically modifying is performed in response to said rule, said first routine is considered to be stalled after a determination that said first routine has been running for at least as long as said interval.

4. (cancelled) ~~A method according to claim 1, wherein said step of automatically modifying comprises:~~

- ~~— adding object code for a timing mechanism to said existing code;~~
- ~~— adding a first instruction to said existing code for said first routine, said first instruction causes a starting of said timing mechanism for said first routine, said first instruction is object code; and~~
- ~~— adding a second instruction to said existing object code for said first routine, said second instruction causes a stopping of said timing mechanism for said first routine, said second instruction is object code.~~

5. (Currently amended) A method according to claim ~~[4]1~~, wherein:

said second instruction is added such that it is executed at all exits of said first routine.

6. (previously presented) A method according to claim 1, wherein:

said first routine is a method; and

said changing said existing code includes adding code to said method to call said new code.

7. (Currently amended) A method according to claim ~~[4]1~~, wherein:

said first routine is a thread.

8. (Currently amended) A method according to claim [4]1, wherein:
said first routine is one of a plurality of routines that comprise a process.

9. (previously presented) A method according to claim 1, wherein said step of using
comprises:

receiving an indication that said first routine has started;
starting a timing mechanism in response to said step of receiving;
receiving an indication that said first routine has completed if said first routine has
completed, said indication is caused by said changed existing code;
stopping said timing mechanism in response to receiving said indication that said first routine
has completed, if said first routine has completed; and
reporting said first routine as stalled if said timing mechanism is not stopped prior to said
timing mechanism being overdue for said first routine.

10. (previously presented) A method for detecting whether a routine has stalled,
comprising:

accessing existing code for a first routine;
automatically modifying said existing code to include new code; and
using said new code to determine if said first routine has stalled, said step of using comprises:
receiving an indication that said first routine has started,
starting a timing mechanism in response to said step of receiving,
receiving an indication that said first routine has completed if said first routine has
completed,
stopping said timing mechanism in response to receiving said indication that said first
routine has completed,
accessing a current time,
verifying that said first routine is not known to be stalled or completed,
accessing a due time,
determining whether due time is earlier than said current time, said timing mechanism

is overdue if said due time is earlier than said current time, and
reporting said first routine as stalled if said timing mechanism is not stopped prior to a
determination that said timing mechanism is overdue.

11. (original) A method according to claim 9, wherein said step of stopping said timing
mechanism comprises the steps of:

determining whether said first routine has been reported as being stalled;
changing said reporting to no longer indicate that said first routine is stalled if said step of
determining concludes that said first routine has been reported as being stalled; and
stopping said timing mechanism if said first routine has not been reported as being stalled.

12. (original) A method according to claim 9, wherein:
said first routine is a thread executing a method, said indication that said first routine has
completed indicates that said thread has exited said method.

13. (cancelled)

14. (previously presented) A method for detecting whether a routine has stalled,
comprising:

receiving an indication that a particular routine has started, said particular routine is one of a
plurality of routines that comprise a process;

starting a timing mechanism for said particular routine in response to said indication that said
particular routine has started;

calculating a due time based on a current time and a time interval;

receiving an indication that said particular routine has completed, if said particular routine
has completed;

determining whether said indication that said particular routine has completed is provided
prior to said due time, said indication that said particular routine has completed is overdue if

provided after said due time; and

reporting said particular routine as stalled if said indication that said particular routine has completed is not provided prior to being overdue.

15. (previously presented) A method according to claim 14, wherein:
said particular routine is a thread;
said plurality of routines are threads; and
at least two or more of said threads, including said particular routine, are run concurrently.

16. (previously presented) A method according to claim 14, wherein:
said particular routine is a method

17. (previously presented) A method according to claim 16, wherein:
said steps of receiving an indication that a particular routine has started and receiving an indication that a particular routine has completed include detecting whether a thread enters said method and does not return within an approximation of an expected time frame.

18. - 21. (cancelled)

22. (previously presented) A method according to claim 14, wherein said determining whether said indication that said first routine has completed is provided prior to said due time comprises:

accessing a new current time;
verifying that said particular routine is not known to be stalled or completed;
accessing said due time; and
determining whether said due time is earlier than said new current time.

23. (cancelled)

24. (previously presented) A method according to claim 14, wherein said step of calculating a due time comprises:

receiving a threshold;

accessing said current time;

determining said due time based on said threshold and said current time; and

adding a indication of said particular routine and said due time to a set of due times for other routines.

25. (previously presented) A method according to claim 24, wherein said step of determining comprises:

determining whether said particular routine has been reported as being stalled;

changing said reporting to no longer indicate that said particular routine is stalled if said particular routine has been reported as being stalled; and

removing said indication of said particular routine and said due time from said set of items if said particular routine has not been reported as being stalled.

26. (original) A method according to claim 25, further comprising the step of:

repeatedly evaluating said set of due times to determine if any of said due times have passed.

27. - 29. (cancelled)

30. (previously presented) A method for detecting whether a routine has stalled, comprising:

receiving an indication that a particular routine has started, said particular routine is one of a plurality of routines that comprise a process;

receiving an indication that said particular routine has completed, if said particular routine has completed; and

reporting said particular routine as stalled if said indication that said particular routine has

completed is not provided prior to being overdue, said particular routine is an instance of a defined routine, said step of reporting includes incrementing a counter that represents a number of instances of said defined routine that are currently stalled and reporting said number of instances of said defined routine that are currently stalled.

31. (previously presented) A method for detecting whether a routine has stalled, comprising:

receiving an indication that a particular routine has started, said particular routine is one of a plurality of routines that comprise a process;

receiving an indication that said particular routine has completed, if said particular routine has completed; and

reporting said particular routine as stalled if said indication that said particular routine has completed is not provided prior to being overdue, said particular routine is an instance of a defined routine, said step of reporting includes determining and reporting how many instances of said defined routine were stalled at a specified time.

32. (previously presented) A method for detecting whether a routine has stalled, comprising:

receiving an indication that a particular routine has started, said particular routine is one of a plurality of routines that comprise a process;

receiving an indication that said particular routine has completed, if said particular routine has completed; and

reporting said particular routine as stalled if said indication that said particular routine has completed is not provided prior to being overdue, said particular routine is an instance of a defined routine, said step of reporting includes receiving a customizable specified time period and reporting how many instances of said defined routine were stalled during said specified time period.

33. - 40. (cancelled)

41. (previously presented) A method according to claim 43, further comprising the step of:

automatically changing existing object code for said method and adding new object code, said new object code performs at least a portion of said step of automatically determining.

42. (cancelled)

43. (previously presented) A method for detecting whether a method has stalled, comprising:

receiving an indication that a particular method of an object is running;

automatically determining whether said method has stalled by detecting whether a thread entered said method and did not return within an approximation of an expected time frame;

determining whether said particular method has been previously reported as being stalled; and

changing said reporting to no longer indicate that said particular method is stalled if it is determined that said particular method has been reported as being stalled and said particular method is no longer stalled.

44. - 47. (cancelled)

48. (previously presented) A method according to claim 50, further comprising the steps of:

accessing a current time;

verifying that said first routine is not known to be stalled or completed;

accessing a due time for said first routine; and

determining whether said due time is earlier than said current time, said timing mechanism is overdue if said step of determining concludes that said first due time is earlier than said current time.

49. (previously presented) A method according to claim 50, wherein said step of stopping

said timing mechanism comprises the steps of:

determining whether said first routine has been reported as being stalled;
changing said reporting to no longer indicate that said first routine is stalled if said first routine has been reported as being stalled; and
stopping said timing mechanism if said first routine has not been reported as being stalled.

50. (previously presented) A method for detecting whether a routine has stalled, comprising the steps of:

receiving an indication that a first routine has started;
starting a timing mechanism in response to said indication that said first routine has started;
receiving an indication that said first routine has completed, if said first routine has completed;
stopping said timing mechanism in response to receiving said indication that said first routine has completed; and
reporting said first routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue, said first routine is an instance of a defined routine, said step of reporting includes incrementing a counter that represents a number of instances of said defined routine that are stalled and reporting said number of instances of said defined routine that are stalled.

51. (previously presented) A method for detecting whether a routine has stalled, comprising the steps of:

receiving an indication that a first routine has started;
starting a timing mechanism in response to said indication that said first routine has started;
receiving an indication that said first routine has completed, if said first routine has completed;
stopping said timing mechanism in response to receiving said indication that said first routine has completed; and
reporting said first routine as stalled if said timing mechanism is not stopped prior to a

determination that said timing mechanism is overdue, said first routine is an instance of a defined routine, said step of reporting includes receiving a customizable specified time period and reporting how many instances of said defined were stalled during said specified time period.

52. (Currently amended) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising:

accessing existing code for a first method;

automatically modifying said existing code for said first method by changing said existing code and adding new code; and

using said new code and said changed existing code to determine if said first method has stalled, wherein said step of automatically modifying comprises,

adding object code for a timing mechanism to said existing code,

adding a first instruction to said existing code for said first method, said first instruction causes a starting of said timing mechanism for said first method, said first instruction is object code, and

adding a second instruction to said existing object code for said first method, said second instruction causes a stopping of said timing mechanism for said first method, said second instruction is object code.

53. - 55. (cancelled)

56. (previously presented) One or more processor readable storage devices according to claim 52, wherein said step of using comprises:

receiving an indication that said first method has started;

starting a timing mechanism in response to said step of receiving;

receiving an indication that said first method has completed, if said first method has completed;

stopping said timing mechanism in response to receiving said indication that said first

method has completed; and

reporting said first method as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue for said first method.

57. (cancelled)

58. (Currently amended) An apparatus, comprising:

one or more storage devices; and

one or more processors in communication with said one or more storage devices, said one or more processors perform a method comprising:

accessing object code for a first routine,

automatically changing said object code and adding new code, and

using said changed object code and said new code to determine if said first routine has stalled, wherein said step of automatically modifying comprises,

adding object code for a timing mechanism to said object code,

adding a first instruction to said object code for said first routine, said first instruction causes a starting of said timing mechanism for said first routine, said first instruction is object code, and

adding a second instruction to said object code for said first routine, said second instruction causes a stopping of said timing mechanism for said first routine, said second instruction is object code.

59. - 61. (cancelled)

62. (original) An apparatus according to claim 58, wherein said step of using comprises the steps of:

receiving an indication that said first routine has started;

starting a timing mechanism in response to said step of receiving;

receiving an indication that said first routine has completed, if said first routine has

completed;

stopping said timing mechanism in response to receiving said indication that said first routine has completed; and

reporting said first routine as stalled if said timing mechanism is not stopped prior to a determination that said timing mechanism is overdue.

63. - 67. (cancelled)

68. (previously presented) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising:

receiving an indication that a particular routine has started;

starting a timing mechanism for said particular routine in response to said indication that said particular routine has started;

calculating a due time based on a current time and a time interval;

receiving an indication that said particular routine has completed, if said particular routine has completed;

determining if said indication that said particular routine has completed is provided prior to said due time, said indication that said particular routine has completed is overdue if provided after said due time; and

reporting said particular routine as stalled if said indication that said particular routine has completed is not provided prior to being overdue.

69. – 70. (cancelled)

71. (previously presented) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising:

receiving an indication that a particular routine has started;
receiving an indication that said particular routine has completed, if said particular routine has completed; and

reporting said particular routine as stalled if said indication that said particular routine has completed is not provided prior to being overdue, said particular routine is an instance of a defined routine, said step of reporting includes incrementing a counter that represents a number of instances of said defined routine that are currently stalled and reporting said number of instances of said defined routine that are currently stalled.

72. (previously presented) An apparatus, comprising:
one or more storage devices; and
one or more processors in communication with said one or more storage devices, said one or more processors perform a method comprising:

receiving an indication that a particular routine is running, said particular routine is one of a plurality of routines that comprise a process,
automatically determining that said particular routine has not stalled,
determining that said particular routine has been previously reported as being stalled,
and
changing said reporting to no longer indicate that said particular routine is stalled.

73. -74. (cancelled)

75. (previously presented) An apparatus according to claim 72, wherein:
said step of automatically determining that said particular routine has not stalled includes detecting that a thread enters a method and returns within an approximation of an expected time frame.

76. (cancelled)

77. (previously presented) An apparatus according to claim 72, wherein said method further comprises the step of:

automatically changing existing object code for said particular routine and adding new object code to said existing object code, said new object code performs said step of automatically determining.

78. -79. (cancelled)

80. (previously presented) One or more processor readable storage devices according to claim 52, wherein:

said existing code is object code; and

said new code is object code.

81. (previously presented) One or more processor readable storage devices according to claim 52, wherein:

said adding new code includes adding object code for a timing mechanism;

said changing said existing code includes adding a first instruction and a second instruction, said first instruction causes a starting of said timing mechanism and said second instruction causes a stopping of said timing mechanism.